UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,740	12/30/2003	Nikolai G. Nikolov	2058.348US1	9016

50400 7590 01/11/2010

SCHWEGMAN, LUNDBERG & WOESSNER/SAP
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER	
TECKLU, ISAAC TUKU	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
01/11/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

uspto@slwip.com
request@slwip.com

Office Action Summary	Application No.	Applicant(s)	
	10/749,740	NIKOLOV, NIKOLAI G.	
	Examiner	Art Unit	
	ISAAC T. TECKLU	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 October 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-10,13-15,22,24,26,28,30,31,43-45 and 47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45, and 47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *10/16/09*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 06/04/2009 has been entered.

2. Claims 2, 11-12, 16-21, 23, 25, 27, 29, 32-42, and 46 have been cancelled.

3. Claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45, and 47 have been examined.

Response to Arguments

4. Applicant's arguments with respect to claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47 have been considered but are moot in view of the new ground(s) of rejection. See of Berry et al. (US 6,742,178 B1) and Avakian et al. (US 7,484,209 B2) below.

Specification

5. The disclosure is objected to because it contains an embedded hyperlink http://jakarta.apache.org/ant on page 2, line 9. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 13-15, and 44 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 13-15 contain the trademarks/trade names JAVA (line 2). Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademarks/trade names are used to identify/describe particular programming languages and, accordingly, the identification/description is indefinite.

Claim 44 recites the limitation "said identities" in line 1. There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, Examiner has treated the above limitation to refer to a method identities.

Claim Rejections - 35 USC § 101

8. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claims 43-45 and 47 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Computer programs/systems claimed as computer listings per se, *i.e.*, the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. *See In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035. Thus, it is essential to distinguish claims that define descriptive material per se from claims that define statutory inventions. See MPEP 2106.

Claim 43 recites a "system" comprising a means for modifying, means for compiling and means for filtering, which can be reasonably interpreted as software, *per se* (see specification ¶ [0053] "… modifier module 352…", ¶ [0109] "… filtering module…", ¶ [0160] "… build tool for compiling…"). Thus, the system is reasonably interpreted as entirely software, which amounts to descriptive material per se. The system is not supported by hardware such as tangible computer storage or execution engine, which would enable one skill in the art to construe that the system, is built from tangible product to carry out any functionality being conveyed from the claim. Thus, the system is software *per se* and therefore is not being tangibly embodied in a manner as to be executable. See MPEP § 2106.01. Accordingly, claims 43-45, and 47 appear to merely set forth descriptive material *per se*, which is nonstatutory.

Claims 44-45 and 47 are rejected for failing to cure the deficiencies of the above rejected non-statutory claim 43 above.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1, 3-5, 7, 9-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berry et al. (US 6,742,178 B1) in view of Avakian (US 7,484,209 B2).

Per claim 1 (Currently Amended), Berry discloses a method comprising:

modifying a classfile after said classfile has been compiled from source code, version of software application said classfile describing properties of a class within an object oriented environment, said modifying comprising (see at least col.2:1-19 "… class files are instrumented with trace hooks… written to a record each time a module is entered or exited…" – Note *compiled Java programs* are typically referred to as *Java class files* – (emphasis added));

modifying a method information structure for each method associated with the software application by adding byte code instructions to said method information structure to cause a plug-in handler method associated with a plug-in handler to execute an output function for each method (see at least col.1:50-60 "… trace tool… log every entry into and every exit from…", col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the

insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…", FIG. 6, FIND TRACE RECORDS MAPPING METHOD NAME TO MAJOR/MINOR CODES 602), modification of the method information structure for each method comprising inserting function calls at entry points and exit points of each method associated with the software application via a bytecode modifier, the method comprising; (e.g. col.2:54-65 "… class file is instrumented with hooks… hook is injected in a method at a critical point in the code… such as where the method will be entered or exited...."), to cause a plug-in handler method associated with a plug-in handler to execute an output function for each method, the plug-in handler to record method information associated with methods (e.g. col.2:54-65 "… class file is instrumented with hooks… hook is injected in a method at a critical point in the code… such as where the method will be entered or exited....when a method is entered or exited, the hooks identifying the entry or exit are also outputted to a trace record…"); and,

compiling results of the modified the class file and providing the results to a graphical user interface to create a graphical representation of the results (see at least col.8:1-10 "… profiling information report with the compiled method mapping file…", e.g. FIG.5, TRACE RECORD

FILE 500), the results including method information (see at least e.g. FIG.5, MAJOR, MINOR, CLASS/METHOD), the method information including dependency hierarchical tree indicting dependency order of the methods (see at least e.g. col.6:15-30 "… methods are identified by major and minor codes in the profiling information report and the HDF file contains mappings between method names and method major and minor codes for the instrumented class…"), and a time hierarchical tree indicting chronological order of the methods (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…"); and

filtering the method information via a filtering module, according to user preferences and the dependency and time hierarchical tree (see at least col.6:30-45 "… if all methods in the report are identified by name and the user is satisfied… the profiling information report must be merged with a more current HDF file… HDF file can the be re-merged and rechecked until the method names are accurate…", col.7:1-15 "… map the method name to the major and minor codes while creating the profiling information report… compiled method mapping file…", col.8:45-60 "… user is then relieved of the tasks of finding and validating the HDF…generating post process profiling information report…", col.9:1-13 "… this mapping information from all records may be compiled into a table for identifying all methods in a class or group of classes…report identifies instrumented methods by the unique names, … understandable for the user…", e.g. FIG. 6, 604).

Berry substantially disclosed the invention as claimed above. Further, Berry discloses method information such as mappings between methods (see at least e.g. col.6:15-30 "… methods are identified by major and minor codes in the profiling information report and the HDF file contains mappings between method names and method major and minor codes for the instrumented

class…"). However, Berry does not explicitly disclose dependency hierarchical tree. Nevertheless, as evidenced by the teaching of Avakian, it is well known to have dependency hierarchical tree for method information (see at least col.30:4-14 "… hierarchical chain of dependency…"). Thus, it is respectfully submitted that it would have been obvious to one skilled in the art at the time the invention was made to have to include dependency hierarchical tree in order to correlate the relationship between parent and child method for consistent profiling, debugging, etc. as once suggested by Avakian (see at least col.30:4-14).

Per claim 3 (Currently Amended), Berry discloses the method of claim [[2]] 1 further comprising:

adding a field information structure to the method, said field information structure describing a field that is to store a numeric identifier of said class (see at least col.6:57-67 "… instrumented method in a class, an entry is added to the class static initializes to output a hook which identified the method…initializes are executed causing hook identifying the method…", see also col.5:60-65 & col.6:1-5).

Per claim 4, Berry discloses the method of claim 3 wherein said numeric identifier is provided to said class by a method of which a dispatch unit is comprised (see at least e.g. FIG. 5, TIMESTAMP, DESCRIPTION 15:985860940 DISPATCH THREAD: e18507a0).

Per claim 5, Berry discloses the method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide an output function treatment in response to an entry point of said method being reached (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode

modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…", FIG. 6, FIND TRACE RECORDS MAPPING MEHTOD NAME TO MAJOR/MINOR CODES 602).

Per claim 7, Berry discloses the classfile modification method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in handler method to provide said output function treatment in response to an exit point of said method being inevitably reached (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…",

FIG. 6, FIND TRACE RECORDS MAPPING MEHTOD NAME TO MAJOR/MINOR CODES 602).

Per claim 9, Berry discloses the method of claim 7 wherein portions of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to any exit point of said method being inevitably reached (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…", FIG. 6, FIND TRACE RECORDS MAPPING MEHTOD NAME TO MAJOR/MINOR CODES 602).

Per claim 10, Berry discloses the method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising during execution of said method (see at least col.6:30-45 "… if all methods in the report are identified by name and the user is satisfied… the profiling information report must be merged with a more current HDF file… HDF file can the be re-merged and rechecked until the method names are accurate…", col.7:1-15

"… map the method name to the major and minor codes while creating the profiling information report… compiled method mapping file…", col.8:45-60 "… user is then relieved of the tasks of finding and validating the HDF…generating post process profiling information report…", col.8:20-30 "… process continue until finalization, or an error in the code occurs…").

Per claim 13, Berry discloses the method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke static instruction (see at least col.2:54-60 "…class file is instrumented with hooks…when class is loaded, the static initializes… identifying the method name and the major and minor codes…", col.7:31-40 "… byte code modification requires the insertion of Java byte code instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, 316).

Per claim 14, Berry discloses the method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke virtual instruction (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…interpretation of existing byte codes…" col.5:60-65, & col.6:1-5 "…major minor timestamp description 12 1 15:985860940 Dispatch thread: e18507a0 22 1 15:985833507 ENTRY: TEST.method_X(I) 12 1 15:985845671 Dispatch thread: e17d5bb0 12 1 15:985860940 Dispatch thread: e1807a0 22 81 15:985833507 EXIT: TEST.method_X_exit 22 2 15:985833507 ENTRY: TEST.method_Y() 22 82 15:985833507 EXIT: TEST.method_Y_exit …").

Per claim 15, Berry discloses the classfile modification method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke

special instruction (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…").

Per claim 22 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 24 (Currently Amended), this is the machine readable medium version of the claimed method discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 26, this is the machine readable storage medium version of the claimed method discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 28, this is the machine readable storage medium version of the claimed method discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 30, this is the machine readable storage medium version of the claimed method discussed above (Claim 9), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 31, this is the machine readable storage medium version of the claimed method discussed above (Claim 10), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 43 (Currently Amended), a system for modifying comprising:

means for modifying a classfile after said classfile has been compiled from a source code version of the software application, said classfile describing properties of a class within an object oriented environment (see at least col.2:1-19 "… class files are instrumented with trace hooks… written to a record each time a module is entered or exited…" – Note *compiled Java programs* are typically referred to as *Java class files* – (emphasis added)), said modifying comprising:

~~means for~~ modifying a method information structure for each method associated with the software application by adding byte code instructions to the said method information structure to cause a plug-in handler method associated with a plug-in handler to execute an output function for said each method (see at least col.1:50-60 "… trace tool… log every entry into and every exit from…", col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point, modification of the method information structure for each method comprising inserting function calls at entry points and exit points of each method associated with the software application

via a bytecode modifier (e.g. col.2:54-65 "… class file is instrumented with hooks… hook is injected in a method at a critical point in the code… such as where the method will be entered or exited...."), to cause a plug-in handler method associated with a plug-in handler to execute an output function for each method, the plug-in handler to record method information associated with methods (e.g. col.2:54-65 "… class file is instrumented with hooks… hook is injected in a method at a critical point in the code… such as where the method will be entered or exited....when a method is entered or exited, the hooks identifying the entry or exit are also outputted to a trace record…"), the system further comprising:

means for compiling results of the ~~modifying of the~~ modified classfile and for providing the results to a graphical user interface to create a graphical representation of the results (see at least col.8:1-10 "… profiling information report with the compiled method mapping file…", e.g. FIG.5, TRACE RECORD FILE 500, Note Apache Ant is well known open source tool used for compiling), the results including method information, the method information including a dependency hierarchical tree indicating dependency order of the methods, and a time hierarchical tree indicting chronological order of the methods (see at least e.g. col.6:15-30 "… methods are identified by major and minor codes in the profiling information report and the HDF file contains mappings between method names and method major and minor codes for the instrumented class…"); and

means for filtering the method information, via a filtering module, according to user preferences and the dependency and time hierarchical trees (see at least col.6:30-45 "… if all methods in the report are identified by name and the user is satisfied… the profiling information report must be merged with a more current HDF file… HDF file can the be re-merged and rechecked until the method names are accurate…", col.7:1-15 "… map the method name to the

major and minor codes while creating the profiling information report… compiled method mapping file…", col.8:45-60 "… user is then relieved of the tasks of finding and validating the HDF…generating post process profiling information report…", col.9:1-13 "… this mapping information from all records may be compiled into a table for identifying all methods in a class or group of classes…report identifies instrumented methods by the unique names, … understandable for the user…", e.g. FIG. 6, 604).

Berry substantially disclosed the invention as claimed above. Further, Berry discloses method information such as mappings between methods (see at least e.g. col.6:15-30 "… methods are identified by major and minor codes in the profiling information report and the HDF file contains mappings between method names and method major and minor codes for the instrumented class…"). However, Berry does not explicitly disclose dependency hierarchical tree. Nevertheless, as evidenced by the teaching of Avakian, it is well known to have dependency hierarchical tree for method information (see at least col.30:4-14 "… hierarchical chain of dependency…"). Thus, it is respectfully submitted that it would have been obvious to one skilled in the art at the time the invention was made to have to include dependency hierarchical tree in order to correlate the relationship between parent and child method for consistent profiling, debugging, etc. as once suggested by Avakian (see at least col.30:4-14).

Per claim 44, wherein said identities are each in a character string format (see at least e.g. col.6:15-30 "… methods are identified by major and minor codes in the profiling information report and the HDF file contains mappings between method names and method major and minor codes

for the instrumented class…").

Per claim 45 (Currently Amended) The system of claim [[44]] 43 wherein further comprising: means for adding a field information structure to the methods, said field information structure describing a field that is to store a numeric identifier of said class (see at least col.6:57-67 "… instrumented method in a class, an entry is added to the class static initializes to output a hook which identified the method…initializes are executed causing hook identifying the method…", see also col.5:60-65 & col.6:1-5).

Per claim 47 The system of claim 43 wherein a portion of said byte code instructions that are added to said method are causing said plug-in handler method to provide said output function treatment in response to an entry point of said method being reached (see at least col.2:54-60 "…class file is instrumented with hooks…", col.7:31-40 "… bytecode modification requires the insertion of Java bytecode instructions… method invocation noted above…", col.7:44-55 "… modified class file is then created… identifying the method…" and e.g. FIG. 3, MORE SELECTED METHODS TO HOOK? 304, LOCATE ENTRY AND EXIT(S) IN METHOD 306, INSERT ENTRY AND EXITS HOOKS FOR MAJOR AND MINOR CODE 308, MAKE MODIFICATIONS TO OTHER COMPONENTS OF THE CLASS FILE…310), the plug-in handler to record method information associated with methods at each entry point and exit point (see at least e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…", FIG. 6, FIND TRACE RECORDS MAPPING MEHTOD NAME TO MAJOR/MINOR CODES 602).

12. Claims 6 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berry et al. (US 6,742,178 B1) in view of Avakian (US 7,484,209 B2), further in view of Bak (US 2003/0093779 A1).

Per claim 6, Berry and Avakian discloses the method of claim 5 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method (see at least Berry col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…");

2) recording an input parameter value for said method (see at least Berry col.1:55-62 "… log every entry into…", col.5:60-65, & col.6:1-5 "…major minor timestamp description 12 1 15:985860940 Dispatch thread: e18507a0 22 1 15:985833507 ENTRY: TEST.method_X(I) 12 1 15:985845671 Dispatch thread: e17d5bb0 12 1 15:985860940 Dispatch thread: e1807a0 22 81 15:985833507 EXIT: TEST.method_X_exit 22 2 15:985833507 ENTRY: TEST.method_Y() 22 82 15:985833507 EXIT: TEST.method_Y_exit …", e.g. FIG. 5, TIMESTAMP, DESCRIPTION 15:985860940 DISPATCH THREAD: e18507a0); and,

Neither Berry nor Avakian explicitly discloses 3) incrementing a counter for said method. Nevertheless as evidenced by the teaching of Bak , incrementing a counter for a method is well known (see at least ¶ [0009]). Thus, it is respectfully submitted that it would have been obvious to one skilled in the art at the time the invention was made to increment a counter for a method to determine which methods to compile using a counter as once suggested by Bak (see at least ¶ [0009]).

Per claim 8, Berry and Avakian disclose the method of claim 7 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method (see at least Berry e.g. col.1:50-60 "… time stamped record is produced…", col.2:5-15 "… class files are instrumented with trace hooks which may be written to a record…");

2) recording an input parameter value for said method (see at least Berry col.1:55-62 "… log every entry into…", col.5:60-65, & col.6:1-5 "…major minor timestamp description 12 1 15:985860940 Dispatch thread: e18507a0 22 1 15:985833507 ENTRY: TEST.method_X(I) 12 1 15:985845671 Dispatch thread: e17d5bb0 12 1 15:985860940 Dispatch thread: e1807a0 22 81 15:985833507 EXIT: TEST.method_X_exit 22 2 15:985833507 ENTRY: TEST.method_Y() 22 82 15:985833507 EXIT: TEST.method_Y_exit …", e.g. FIG. 5, TIMESTAMP, DESCRIPTION 15:985860940 DISPATCH THREAD: e18507a0, e.g. FIG. 5, TIMESTAMP, DESCRIPTION 15:985860940 DISPATCH THREAD: e18507a0);

Neither Berry nor Avakian explicitly disclose 3) incrementing a counter for said method. Nevertheless as evidenced by the teaching of Bak, incrementing a counter for a method is well known (see at least ¶ [0009]). Thus, it is respectfully submitted that it would have been obvious to one skilled in the art at the time the invention was made to increment a counter for a method to determine which methods to compile using a counter as once suggested by Bak (see at least ¶ [0009]).

Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192